

CLAIMS

1. A graphics controller comprising:
a bin pointer list containing information regarding image data for each of a plurality of subscenes; and
a pointer cache system to store accesses to the bin pointer list.
2. The graphics controller of claim 1, wherein said pointer cache system comprises a cache tag array section and a cache data array section.
3. The graphics controller of claim 2, wherein each cache tag in said cache tag array section corresponds to at least four pieces of cache data in said cache data array section.
4. The graphics controller of claim 2, wherein said cache tag array section receives a data request and determines if said cache data array contains information related to said data request.
5. The graphics controller of claim 4, wherein said pointer cache system further comprises a decoupling section to create a non-blocking pointer cache system.
6. The graphics controller of claim 5, wherein if said cache data array contains information related to said data request then said data request is placed in

said decoupling section.

7. The graphics controller of claim 5, wherein if said cache data array does not contain information related to said data request then a data request is made for the missing data and the data request is placed in said decoupling section.

8. The graphics controller of claim 5, wherein said pointer cache system performs a recovery operation following an out of memory condition.

9. The graphics controller of claim 8, wherein said recovery operation comprises blocking access to the cache data array section, creating a new bin pointer list, examining entries in said decoupling section and subsequently unblocking access to the cache data array section.

10. The graphics controller of claim 1, wherein said graphic controller vertically arranges said subscenes.

11. A computer graphics system comprising:
a memory to store image data;
a bin pointer list to store information regarding a plurality of image subscenes;
and
a pointer cache system to maintain data regarding said plurality of image subscenes, wherein said pointer cache system comprises a tag array section and a

~~data array section.~~

12. The computer graphics system of claim 11, wherein each cache tag in said tag array section corresponds to four pieces of data in said data array section.

13. The computer graphics system of claim 11, wherein said tag array section receives a data request and determines if said data array contains information related to said data request.

14. The computer graphics system of claim 13, wherein said pointer cache system further comprises a decoupling section to create a non-blocking pointer cache system.

15. The computer graphics system of claim 14, wherein if said data array section contains information related to said data request then said data request is placed in said decoupling section.

16. The computer graphics system of claim 14, wherein if said data array does not contain information related to said data request then a data request is made for the missing data and the data request is placed in said decoupling section.

17. The computer graphics system of claim 14, wherein said pointer cache system performs a recovery operation following an out of memory condition.

18. The computer graphics system of claim 17, wherein said recovery operation comprises blocking access to the data array section, creating a new bin pointer list, examining entries in said decoupling section and subsequently unblocking access to the data array section.

19. The computer graphics system of claim 11, wherein said graphic controller vertically arranges said subscenes.

20. A method comprising:
maintaining a bin pointer list containing information regarding image data for a plurality of subscenes;
making a data request to a pointer cache system to determine if said pointer cache system contains information relating to said data request; and
obtaining information from said pointer cache system.

21. The method of claim 20, further comprising obtaining said information from a separate memory prior to obtaining said information from said pointer cache system if said pointer cache system does not contain information relating to said data request.

22. The method of claim 20, wherein said pointer cache system is a non-blocking cache system.

SIRIUS

23. The method of claim 20, further comprising performing a recovery operation following an out of memory condition.

24. The method of claim 20, wherein said pointer cache system comprise a cache tag array section and a cache data array section.

2000 1000 800 600 400 200